

Données, structure de code et fonctions

Objectifs

Très rapide. C'est un simple rappel sur les types de données, les structures de code et les fonctions.

Aucune notion de DOM pour l'instant. On exécute un fichier javascript dans le terminal intégré à WebStorm.

Le thème de Star Wars est utilisé pour essayer d'attirer l'attention sur une démonstration qui peut sembler inutile pour les apprenants.

Mise en place

- Ouvrir Webstorm
- Créer un nouveau document javascript : **New** > **JavaScript File**

Type de données

On va faire la démonstration de ce qui a été dit pendant le powerpoint, avec des données extrêmement simple.

- Deux chaînes de caractères
- Un objet
- Un tableau

```
console.group("Type de données : ");
// Déclarations :
let jedi = "Obi-Wan Kenobi";
let sith = "Anakin Skywalker";
let aujourd'hui = new Date();
let planetes = ["Tatooïne", "Dagobah", "Naboo", "Aldérande"];
// Vérifications des types de données :
console.log(typeof jedi);
console.log(typeof sith);
console.dir(aujourd'hui);
console.table(planetes);
console.groupEnd();
```

On en profite pour introduire *console.dir()* et *console.table()* pour que la démonstration ne soit pas ennuyeuse.



Expliquer *typeof* qui prendra tout son sens dans le module 02 lorsque l'on fera de l'objet.

Structure de code

- Une boucle `for...of` va permettre de faire une boucle sur le tableau de planètes.
- Une condition permet de faire passer Anakin du côté obscur de la force.

Plus précisément, si la date du jour est paire, rien ne bouge. Sinon, il change de nom.

```
console.group("Structures de code : ");
console.log("Les planètes avec une boucle for..of : ");
for (const planete of planetes) {
    console.dir(planete);
}
console.log("Une condition :")
if (aujourd'hui.getDay() % 2) {
    sith = "Dark Vador";
}
console.dir(sith);
console.groupEnd();
```

Fonctions

Ici, on va utiliser une fonction pour aller récupérer dans un tableau un élément bien précis passé en paramètre.

```
/**
 * Permet de retrouver un principe du code jedi
 * par son index
 *
 * @param index
 * @returns {string}
 */
function code_jedi(index) {
    const principes = [
        "Il n'y a pas d'émotion, il y a la paix",
        "Il n'y a pas d'ignorance, il y a la connaissance",
        "Il n'y a pas de passion, il y a la sérénité",
        "Il n'y a pas de chaos, il y a l'harmonie",
        "Il n'y a pas la mort, il y a la Force"
    ]
    return principes[index];
}
console.log(code_jedi(1));
```

Insister sur les bonnes pratiques de développement :

- L'indentation
- Les commentaires au-dessus de la fonction



C'est le bon moment pour parler de la portée des variables.